Theoretical Computer Science 79 (1991) 195-207
North-Holland

Computation, on Macsyma, of the minimal differential representation of noncommutative polynomials

N.E. Oussous

Laboratoire d'Informatique Fondamentale de Lille, U.A. 369 du C.N.R.S. Université de Lille I, 59655 Villeneuve d'Ascq Cedex, France

Abstract

Oussous, N.E., Computation, on Macsyma, of the minimal differential representation of noncommutative polynomials, Theoretical Computer Science 79 (1991) 195-207.

We present here a package of *Macsyma* programs, allowing the manipulation of words, and noncommutative power series over a finite alphabet. On the basis of the works of Fliess and Reutenauer concerning the local minimal realization of analytical systems, we present an algorithm allowing the computation of the local minimal realization of finite generating power series. We describe that algorithm in the algebraic computation language, *Macsyma*.

1. Introduction

The realization problem has been studied by many authors. We quote [15] for realization of linear systems, [2, 4, 9] for realization of particular nonlinear systems called bilinear one and [14, 22] for realization of general nonlinear systems which have regular solutions for all time and commands. Each author has used tools more or less difficult to prove existence and uniqueness of realization.

Fliess [8] and Reutenauer [20] have used noncommutative formal power series as a principal tool to prove existence and uniqueness of the local and minimal realization of nonlinear analytical dynamic systems. However, their results do not give in general a complete effective computation of the minimal realization.

First, we present some tools to manipulate noncommutative formal power series (these tools are studied and implemented in [10, 11]. After that, we give the definitions and some properties of Lyndon words [3, 16, 18] and Lyndon basis of free-Lie-algebra $\mathscr{Lie}\langle X\rangle$ [23].

In the last paragraph, we recall essential results on the local realization of nonlinear dynamical systems. We show that by using the Chen–Fox–Lyndon basis of $\mathscr{Lie}\langle X\rangle$ and the fact that the set of Lyndon words is a transcendance basis of noncommutative polynomial algebra with Shuffle product [18, 19], we can compute the local minimal realization of nonlinear dynamical systems of which generating power series are finite. This realization has been implemented with the computer algebra system *Macsyma* [17] in [11, 12].

Finally we describe the complete algorithm in this particular case, and a detailed example of its execution.

2. Noncommutative formal power series

2.1. Alphabet and words

Let $X=\{x_0, x_1, \ldots, x_{m-1}\}$ be a finite set of *letters* called an *alphabet*. We denote by X^* the free monoid generated by X. An element of X^* is called a *word*. The empty word, denoted by ε, is the word which contain no letters. We denote by $|w|$ the *length* of the word w. That is the number of letters in the word.

2.2. Formal power series

Recall that a formal power series S is a *mapping* from the free monoid X^* into $\mathbb{R}$ which associates to each word w its *coefficient* $S(w)$ denoted by $\langle S \mid w\rangle$. The formal series S will also be denoted by the formal sum

$$S=\sum_{w\in X^*}\langle S\mid w\rangle w.$$

The set of all formal power series is denoted by $\mathbb{R}\langle\langle X\rangle\rangle$ and is an algebra for the *Cauchy product*.

The *order* of the power series S is denoted by $\omega(S)$ and defined [1, 10] by:

$$\omega(S)=\begin{cases}\inf\{|w| \mid \langle S\mid w\rangle\neq 0\} & \text{if } S\neq 0,\\ +\infty & \text{if } S=0.\end{cases}$$

The *support* of the formal power series S is the language

$$\operatorname{Supp}(S)=\{w\in X^* \mid \langle S\mid w\rangle\neq 0\}.$$

The power series, without constant term is called *proper*.

A *polynomial* is a formal power series that has a *finite support*. The set of all polynomials is a sub-algebra of $\mathbb{R}\langle\langle X\rangle\rangle$ denoted by $\mathbb{R}\langle X\rangle$.

The *degree* of a polynomial P is defined as follows:

$$\deg(P)=\begin{cases}\sup\{|w| \text{ with } w\in\operatorname{supp}(P)\} & \text{if } P\neq 0,\\ -\infty & \text{if } P=0.\end{cases}$$

We denote $\mathscr{L}ie\langle X\rangle$ the *free-Lie-algebra* generated by X in which the *Lie-brackets* are defined by

$$[x_i, x_j] = x_i x_j - x_j x_i.$$

Lie-polynomials are defined by the following grammar:

LiePoly::= LieMon | LieMon + LiePoly,

LieMon::= LieWord | Coeff * LieWord,

LieWord::= Letter | [LieWord, LieWord],

Letter::= $x_0 | x_1 | \ldots | x_{m-1}$,

Coeff::= Real.

2.3. *Operations over power series*

2.3.1. *Shuffle product*

The shuffle product of two words [7, 10] is defined recursively as follows:

$$\begin{cases} \forall w \in X^*, & w ш \varepsilon = \varepsilon ш w = w, \\ \forall u, v \in X^*, \forall x, y \in X, & (xu) ш (yv) = x(u ш (yv)) + y((xu) ш v). \end{cases}$$

This definition of shuffle product can be easily implemented, but it is expensive in CPU time and it takes up more memory space.

This product is commutative and associative, and can be extended to formal power series by setting for S and T belonging in $\mathbb{R}\langle\langle X\rangle\rangle$:

$$S ш T = \sum_{u,v \in X^*} \langle S|u\rangle\langle T|v\rangle u ш v.$$

2.3.2. *Remainder's computation*

Definition 2.1. Let $S \in \mathbb{R}\langle\langle X\rangle\rangle$ be a formal power series. We set, for $u \in X^*$,

$$u \lhd S = \sum_{w \in X^*} \langle S|wu\rangle w \quad \left(\text{resp. } S \rhd u = \sum_{w \in X^*} \langle S|uw\rangle w\right),$$

which will be called the *right* (resp. *left*) remainder of the power series S by the word u.

In other words, we have, for $w \in X^*$,

$$\langle u \lhd S|w\rangle = \langle S|wu\rangle \quad \text{and} \quad \langle S \rhd u|w\rangle = \langle S|uw\rangle. \tag{1}$$

Example 2.2. Let $X = \{x_0, x_1\}$, $S = x_0x_1x_0^2 + 2x_1x_0^2$, and $u = x_1x_0$, then $u \lhd S = 0$ and $S \rhd u = 2$.

Remark 2.3. (1) $\forall S \in \mathbb{R}\langle\langle X\rangle\rangle$, $\varepsilon \lhd S = S \rhd \varepsilon = S$.

(2) Under the definition of support, we can easily see that

$$\mathrm{Supp}(u \lhd S) = \{w \in X^* \mid wu \in \mathrm{Supp}(S)\},$$

$$\mathrm{Supp}(S \rhd u) = \{w \in X^* \mid uw \in \mathrm{Supp}(S)\}.$$

2.4. Hankel and Lie-Hankel matrix

Definition 2.4 (Fliess [6]). Let $S \in \mathbb{R}\langle\langle X\rangle\rangle$ be a formal power series. The Hankel matrix associated to S is an infinite array, denoted by $\mathcal{H}_S$, of which lines and columns are indexed by the elements of X^*, such that

$$\mathcal{H}_S(u, v) = \langle S|uv\rangle = \langle S \rhd u|v\rangle.$$

Definition 2.5 (Fliess [6], Berstel and Reutenauer [1]). The Hankel rank of the formal power series S, denoted by $\mathcal{HR}_S$, is the rank of the Hankel matrix $\mathcal{H}_S$, if it is finite and $+\infty$ elsewhere.

The Hankel rank is usually infinite. When this rank is finite, the formal power series is said to be *rational* [1, 4, 9].

The rows (resp. columns) of $\mathcal{H}_S$ generate a vector space which has a dimension equal to the rank of $\mathcal{H}_S$. We denote by $\mathcal{L}_u$ the row of $\mathcal{H}_S$ indexed by the word u. We have [10]

$$\forall w \in X^*, \quad \mathcal{L}_u(w) = \langle S|uw\rangle = \langle S \rhd u|w\rangle,$$

hence $\mathcal{L}_u = S \rhd u$.

The *row* $\mathcal{L}_u$ of $\mathcal{H}_S$ contain exactly the components of the *right remainder* $S \rhd u$ of S by u. So, we have

$$\mathcal{HR}_S = \dim[\operatorname{span}\{\mathcal{L}_u \mid u \in X^*\}].$$

Let $P \in \mathbb{R}\langle X\rangle$. We define the row $\mathcal{L}_P$ of $\mathcal{H}_S$ by

$$\mathcal{L}_P = (\langle S|Pu\rangle)_{u \in X^*} = (\langle S \rhd P|u\rangle)_{u \in X^*}.$$

This row contains the components of the right remainder $S \rhd P$, and we have

$$\mathcal{L}_P = \sum_{v \in X^*} \langle P|v\rangle \mathcal{L}_v.$$

Remark 2.6. All notions that we stated for $\mathcal{H}_S$'s rows can be stated for $\mathcal{H}_S$'s columns and left remainders of S.

Definition 2.7 (Jacob and Oussous [12], Reutenauer [20]). Let $S \in \mathbb{R}\langle\langle X\rangle\rangle$. We define the Lie-Hankel matrix associated to S as an infinite array, denoted $\mathcal{LH}_S$, of which the lines are indexed by some totally ordered basis of $\mathcal{L}ie\langle X\rangle$ and the columns are indexed by X^* (sorted for lexicographical by length order) such that

$$\mathcal{LH}_S(P_i, w) = \langle S|P_i w\rangle = \langle S \rhd P_i|w\rangle.$$

Definition 2.8 (Reutenauer [20]). The Lie-rank of the power series S, denoted by $\mathcal{LR}_S$, is the rank of the Lie-Hankel matrix $\mathcal{LH}_S$ if it is finite and $+\infty$ elsewhere.

According to [8], we have

$$\mathcal{LR}_S = \dim S \rhd \mathcal{L}ie\langle X\rangle = \dim \operatorname{span}\{\mathcal{L}_P \mid P \in \mathcal{L}ie\langle X\rangle\}.$$

We show easily that $\mathcal{LR}_S = \operatorname{Rank}(\mathcal{LH}_S)$.

3. Some recalls over Lyndon words

3.1. Lexicographical order

Let X be a finite and totally ordered alphabet, and X^* the free monoid generated by X. The *usual lexicographical order* on X^* [16] is defined as follows: $\forall u, v \in X^*$, $u < v$ if and only if,

(i) $\exists w \neq \varepsilon$ such that $uw = v$, or

(ii) $\exists x, y, z \in X^*$ and $a, b \in X$ such that $u = xay$, $v = xbz$ and $a < b$.

With this order, we have the following properties:

(1) $\forall w \in X^*$, $u < v \Leftrightarrow wu < wv$.

(2) If $v \notin uX^*$, $\forall w, z \in X^*$, $u < v \Rightarrow uw < vz$.

3.2. Conjugation classes

A word u is called to be a *factor* of a word v if

$$\exists x, y \in X^* \text{ such that } v = xuy.$$

If $x = \varepsilon$ (resp. $y = \varepsilon$), we say that u is a *left* (resp. *right*) *factor* of v, *proper* if $y \neq \varepsilon$ (resp. $x \neq \varepsilon$).

Definition 3.1 (Lothaire [16], Melançon and Reutenauer [18]). Two words u and v are said to be *conjugate* if

$$\exists x, y \in X^* \text{ such that } u = xy \text{ and } v = yx.$$

3.3. Lyndon words and Lyndon basis construction

3.3.1. Lyndon words

Details can be found in [3, 16, 18].

Definition 3.2. A word $w \in X^*$ is a Lyndon word if and only if it satisfies one of the equivalent following statements:

(i) it is strictly smaller than any of its conjugates,

(ii) it is strictly smaller than any of its proper right factors.

Let L denote the set of Lyndon words over X.

Example 3.3. $X = \{x_0, x_1\}$, x_0, x_1, x_0x_1, $x_0^2x_1$, $x_0x_1^2, \ldots$, are Lyndon words over X.

Properties 3.4. (1) Let $w \in L\backslash X$ and m its *longest proper right factor* in L. If $w = lm$, then $l \in L$ and $l < lm < m$. The couple $\sigma(w) = (l, m)$ is called the *standard factorization* of w.

$$\sigma(x_0^2x_1^2) = (x_0, x_0x_1^2) \neq (x_0^2x_1, x_1).$$

(2) $w \in L$ if and only if

either $w \in X$,

or $w = lm$ with $l, m \in L$ and $l < m$.

The last property gives us an algorithm to construct the Lyndon words up to a given degree.

3.3.2. Lyndon basis

For more precise details, see [12, 16, 18, 23]. Under Lyndon words, we can construct *Lyndon Basis* (called also *Chen-Fox-Lyndon basis*) which is defined recursively as follows:

$$c(x) = x \quad \text{for } x \in X,$$

$$c(w) = [c(l), c(m)] \quad \text{if } w \in L \backslash X \text{ and } \sigma(w) = (l, m).$$

where the brackets are Lie brackets. This definition gives us an algorithm to construct this basis.

One advantage of this basis is that by the same algorithm we can produce Lyndon words and associated elements of the Lyndon basis of $\mathscr{L}ie\langle X\rangle$.

Example 3.5. The following Lyndon basis elements are associated respectively with Lyndon words given in the last example:

$$x_0, x_1, [x_0, x_1], [x_0, [x_0, x_1]], [[x_0, x_1], x_1], \ldots .$$

On L, we can define many total order. In particular, we can define the lexicographical and the lexocographical order by length. Under the definition of the Lyndon basis, all order defined on L induced an order on Lyndon basis.

If $(P_i)_{i\geqslant 1}$ is the Lyndon basis of $\mathscr{L}ie\langle X\rangle$, and if $(l_j)_{j\geqslant 1}$ is the Lyndon words associated with this basis, then we have the following important relation:

$$P_i = l_i + \sum_j \alpha_j m_j, \quad m_j > l_i \text{ and } \alpha_j \in \mathbb{Z}.$$

Hence, we have

$$\langle l_j | p_i\rangle = \langle l_j \rhd P_i | \varepsilon\rangle = \delta_{ij} \tag{2}$$

Example 3.6

$$\begin{aligned} P_i &= [x_0, [x_0, x_1]] \\ &= x_0[x_0, x_1] - [x_0, x_1]x_0 \\ &= x_0^2 x_1 - 2x_0x_1x_0 + x_1x_0^2 \\ &= l_i + (-2x_0x_1x_0 + x_1x_0^2). \end{aligned}$$

4. Nonlinear dynamical systems

4.1. Definition

We consider a system of the following form:

$$(\Sigma) \qquad \begin{cases} \dot{q}(t) = \sum_{i=0}^{m-1} u_i(t) Y_i(q) \text{ with } u_0(t) \equiv 1, \\ y(t) = h(q(t)). \end{cases}$$

where q belongs to a *connected* $\mathbb{R}$*-analytic variety* Q, the Y_i are *analytic vector fields*, h is an $\mathbb{R}$*-analytic function* called *observation*, defined in the neighbourhood of the given *initial state* $q(0)$, and the inputs $u_1, \ldots, u_{m-1}$ are *real and piecewise continuous*.

We set $u = (u_0, u_1, \ldots, u_{m-1})$ which will be called the input function.

4.2. Fundamental formula

Each *input* u_i, is associated with the *letter* x_i $(0 \leq i \leq m-1)$. The set of all letters, $X = \{x_0, x_1, \ldots, x_{m-1}\}$, will be called the *command alphabet*. Let X^* be the free monoid generated by X.

For each word $w \in X^*$, we denote by Y_w the differential operator defined as follows:

$$Y_\varepsilon = \text{Identity},$$

$$Y_w = Y_{i_1} \circ Y_{i_2} \circ \cdots \circ Y_{i_k} \quad \text{if } w = x_{i_1} x_{i_2} \ldots x_{i_k}.$$

The *action* of the differential operator Y_w over the analytic function f defined on the variety Q, is denoted by $Y_w \circ f$.

For small enough time and inputs, the output y of the system (Σ) is given by the *Peano-Baker-Formula* [8, 20], also called the *Fliess fundamental formula*:

$$y(t) = \sum_{w \in X^*} (Y_w \circ h)|_{q(0)} \int_0^t \delta_u w \tag{3}$$

where $|_{q(0)}$ means evaluation in $q(0)$, $\int_0^t \delta_u w$ is the *iterated integral* defined recursively as follows:

$$\text{if } w = \varepsilon, \text{ then } \int_0^t \delta_u \varepsilon = 1,$$

$$\text{if } w = x_i \in X, \text{ then } \int_0^t \delta_u x_i = \int_0^t u_i(\tau)\, d\tau,$$

$$\text{if } w = v x_i, \text{ then } \int_0^t \delta_u w = \int_0^t \left(\int_0^\tau \delta_u v \right) u_i(\tau)\, d\tau.$$

This definition is symmetric to that in [7]. In particular, since $u_0(\tau) \equiv 1$,

$$\int_0^t \delta_u x_0 = \int_0^t u_0(\tau)\, d\tau = t.$$

4.3. Generating power series and vector fields

The Input/Output behaviour of system (Σ) is completely defined by its *generating power series* g in the noncommutative variables $x_0, x_1, \ldots, x_{m-1}$, given by the formula [8]

$$g = \sum_{w \in X^*} \langle g | w \rangle w = \sum_{w \in X^*} (Y_w \circ h)|_{q(0)} w. \tag{4}$$

Thus, the output $y(t)$, given by (3), can be written

$$y(t)=\sum_{w\in X^*}\langle g|w\rangle\int_0^t \delta_u w. \tag{5}$$

The variety Q is of dimension N. Let $z_1, z_2, \ldots, z_N$ be a system of *local coordinates*. The $\mathbb{R}$-algebras of formal series and polynomials in *commutative variables* $z_1, z_2, \ldots, z_N$ are denoted respectively by $\mathbb{R}[\![z_1, z_2, \ldots, z_N]\!]$ and $\mathbb{R}[z_1, z_2, \ldots, z_N]$. The *vector fields* Y_i defined above can be written

$$Y_i=\sum_{k=1}^{N}\theta_i^k(z)\frac{\partial}{\partial z_k},\qquad \theta_i^k(z)\in\mathbb{R}[\![z_1, z_2, \ldots, z_N]\!]. \tag{6}$$

Thus, let f be an analytic function defined over the variety Q, then the action of the vector fields Y_i over the function f can be written

$$Y_i\circ f=\sum_{k=1}^{N}\theta_i^k(z)\frac{\partial f}{\partial z_k}. \tag{7}$$

5. Realization of nonlinear dynamical systems

Locally, the realization problem can be expressed as follows [8]:

> Let an Input/Output behaviour be given by its generating power series: is there a differential system like (Σ) which has the same generating power series? Describe it, in the positive case.

5.1. Power series produced differentially

Definition 5.1 (Fliess [8]). The formal power series $g\in\mathbb{R}\langle\!\langle X\rangle\!\rangle$ is produced differentially if and only if, there exist:

(1) an integer $r\in\mathbb{N}$,

(2) a homomorphism $\mathcal{Y}$ from X^* into differential operator algebra over $\mathbb{R}[\![z_1, \ldots, z_r]\!]$ such that $\forall x_i\in X$, $Y_i=\mathcal{Y}(x_i)$ is a vector field,

(3) a commutative power series $h\in\mathbb{R}[\![z_1, \ldots, z_r]\!]$,

such that

$$\forall w\in X^*,\quad \langle g|w\rangle=(\mathcal{Y}(w)\circ h)|_0, \tag{8}$$

where $|_0$ means evaluation in $z_1=\cdots=z_r=0$.

The couple $(\mathcal{Y}, h)$ is called *differential representation* of g, of dimension r. From (4) and (8), it is obvious that

> g is the generating power series of a system like (Σ)
> if and only if
> g is produced differentially.

Thus

The study of local realization is equivalent to
the study of differential representations.

5.1.1. Fliess theorem

Theorem 5.2 (Fliess [8]). *The power series $g \in \mathbb{R}\langle\langle X\rangle\rangle$ is produced differentially if and only if its Lie-rank, r, is finite. In this case, r is equal to the smallest dimension of all its differential representations. If $(\mathcal{Y}, h)$ and $(\mathcal{Y}', h')$ are two differential representations of dimension r of g, then there exists a continuous automorphism φ of $\mathbb{R}[[z_1, \ldots, z_r]]$ such that*

$$\forall w \in X^*, \forall k \in \mathbb{R}[[z_1, \ldots, z_r]], \quad h' = \varphi(h) \text{ and } \varphi(Y_w \circ k) = Y'_w \circ \varphi(k).$$

Y_w (resp. Y'_w) means the image of w by $\mathcal{Y}$ (resp. $\mathcal{Y}'$).

The realization $(\mathcal{Y}, h)$, unique up to isomorphism, is said to be *minimal* or *reduced.*

6. The realization algorithm

We consider the Lie-sub-algebra $\mathcal{A}(g)$ of $\mathcal{L}ie\langle X\rangle$ of dimension r defined by

$$\mathcal{A}(g) = \{P \in \mathcal{L}ie\langle X\rangle \mid g \rhd P = 0\}.$$

$\mathcal{A}(g)$ is generated by the Lie-polynomials that annul the power series g.

We set

$$\mathcal{V}(\mathcal{A}(g)) = \{Q \in \mathbb{R}\langle\langle X\rangle\rangle \mid Q \rhd \mathcal{A}(g) = 0\}.$$

According to [20], the realization consists of

(1) Find some Lie polynomials $P_1, \ldots, P_r$ such that $g \rhd P_1, \ldots, g \rhd P_r$ is a basis of the vector space $g \rhd \mathcal{L}ie\langle X\rangle$, that is a basis of $\mathcal{L}ie\langle X\rangle$ modulo $\mathcal{A}(g)$.

(2) Find the formal power series Z_j without constant term such that

$$\begin{array}{ll} \text{(i)} & \langle Z_j \rhd P_i | \varepsilon\rangle = \delta_{ij} \text{ for } i \leq r, \\ \text{(ii)} & Z_j \in \mathcal{V}(\mathcal{A}(g)), \end{array} \tag{9}$$

where $\langle Z_j \rhd P_i | \varepsilon\rangle = \langle Z_j | P_i\rangle$.

(3) Express g as a commutative series on the Z_j for the shuffle product

$$g = \sum_\alpha c_\alpha Z^{\sqcup\!\sqcup\, \alpha}.$$

(4) Compute the series $x_i \lhd Z_j$ and express these as commutative series on Z_j for the shuffle product.

(5) Translate (directly) these expressions as giving a differential production of g.

Here, we restrict the problem to polynomial generating series in order to obtain a complete computer algorithm.

We choose the Chen–Fox–Lyndon-basis, defined above, as the Lie-basis. This basis and the set of Lyndon words are sorted lexicographically by length order. Thus, for a polynomial, we obtain all the lines of the Lie-Hankel matrix.

We know, according to [18, 19], that the Lyndon words are a transcendence basis of the shuffle algebra $\mathbb{R}\langle X\rangle$. Elsewhere, we have the important relation (2).

The realization construction is based on the construction of the polynomials $(Z_i)_{1\le i\le r}$. Those polynomials will be built as a linear combination of shuffles of Lyndon words [13].

We use the algorithm shown in Fig. 1 to express the polynomials g as commutative polynomials on the Z_i for the shuffle product. We obtain the expression of the observation h. We use the same algorithm to compute the components of the vector fields.

The algorithm has four parameters:

- g: the generating power series,
- Z: the set of polynomials Z_j,
- LP: the basis of $\mathscr{Lie}\langle X\rangle$ modulo $\mathscr{A}(g)$,
- r: the Lie-rank of g,

and gives as a result the expression G of g as a linear combination on the polynomials Z_j for the shuffle; and the observation h as a commutative polynomial on z_j.

```
Express(g,Z,LP,r)
  ◇ T ← S,  H ← 0,  j ← r,
  ◇ fact1 ← fact ← 1,
  ◇ WHILE (j ≠ 0 and S ≠ 0) DO
    • n ← 0,
    • WHILE T▷Pj ≠ 0 DO
        o T ← T▷Pj,
        o n ← n+1,
    • ENDWHILE
    • cst ← constant term of T,
    • IF cste ≠ 0 THEN
        o S ← S − cst/n! * (fact1 ш qj^шn),
        o H ← H + cst/n! * (fact * qj^n),
        o fact1 ← fact ← 1,
        o T ← S,
        o j ← j+k,  k ← 0,
    • ELSE
        o fact1 ← 1/n! * (fact1 ш qj^шn),
        o fact ← 1/n! * (fact * qj^n),
        o j ← j−1,  k ← k+1,
  ◇ ENDWHILE
```

Fig. 1. Algorithm to compute observation and components of vector fields.

To compute the component θ_i^j of the vector fields Y_i, we use the same algorithm with $x_i \lhd Z_j$ as the first argument.

7. Example

Let $g = x_1x_0x_1 + x_0x_1x_0x_1$, $X = \{x_0, x_1\}$. The $\mathscr{L}ie\langle X\rangle$-basis up to degree 4 is

$$\begin{array}{llll} P_1 = x_0, & P_3 = [x_0, x_1], & P_4 = [x_0, [x_0, x_1]], & P_6 = [x_0, [x_0, [x_0, x_1]]], \\ P_2 = x_1, & & P_5 = [[x_0, x_1], x_1], & P_7 = [x_0, [[x_0, x_1], x_1]], \\ & & & P_8 = [[[x_0, x_1], x_1], x_1]. \end{array}$$

The corresponding Lyndon words are

$$\begin{array}{llll} l_1 = x_0, & l_3 = x_0x_1, & l_4 = x_0^2x_1, & l_6 = x_0^3x_1, \\ l_2 = x_1, & & l_5 = x_0x_1^2, & l_7 = x_0^2x_1^2, \\ & & & l_8 = x_0x_1^3. \end{array}$$

The right remainders of g by the elements of $\mathscr{L}ie\langle X\rangle$-basis are

$$\begin{array}{llll} g \rhd P_1 = x_1x_0x_1, & g \rhd P_3 = -x_1 + x_0x_1, & g \rhd P_4 = -2x_1, & g \rhd P_6 = 0, \\ g \rhd P_2 = x_0x_1, & & g \rhd P_5 = -2, & g \rhd P_7 = -2, \\ & & & g \rhd P_8 = 0. \end{array}$$

The Lie-Hankel matrix is

$$\begin{array}{r} \\ x_0 \\ x_1 \\ [x_0, x_1] \\ [x_0, [x_0, x_1]] \\ [[x_0, x_1], x_1] \\ [x_0, [[x_0, x_1], x_1]] \end{array} \begin{array}{c} \begin{array}{cccc} \varepsilon & x_1 & x_0x_1 & x_1x_0x_1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

The Lie-rank of this polynomial is $\mathscr{L}\mathscr{R}_g = 4$.

The dependence relations are

$$F_1 = P_4 - 2P_3 + 2P_2,$$

$$F_2 = P_7 - P_5.$$

$$\mathscr{A}(g) = \text{span}(\{F_1, F_2, P_6, P_8\} \cup \{P \in \mathscr{L}ie\langle X\rangle \mid \deg(P) > 4\}).$$

We consider the duals of relations F_1, F_2.

The linear map associated to F_i is

$$F = \begin{pmatrix} \langle F_1 | \cdot \rangle \\ \langle F_2 | \cdot \rangle \end{pmatrix} : \mathbb{R}^6 \to \mathbb{R}^{6-4},$$

$$x = (x_1, x_2, \ldots, x_6) \mapsto F(x).$$

$$MF = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 2 & -2 & 1 & 0 & 0 \end{pmatrix}$$

$\dim\ker(F)=4$. The $\ker(F)$-basis is

$$v_1=\begin{pmatrix}1\\0\\0\\0\\0\\0\end{pmatrix},\qquad v_2=\begin{pmatrix}0\\1\\1\\0\\0\\0\end{pmatrix},\qquad v_3=\begin{pmatrix}0\\-\frac{1}{2}\\0\\1\\0\\0\end{pmatrix},\qquad v_4=\begin{pmatrix}0\\0\\0\\0\\1\\1\end{pmatrix}.$$

The pseudo-coordinates are

$$m_i=(l_1,l_2,l_3,l_4,l_5,l_7)\,.\,v_i,\quad 1\leqslant i\leqslant 4.$$

$$\begin{aligned}
&m_1=l_1=x_0, && P_1=x_0,\\
&m_2=l_3+l_2=x_0x_1+x_1, && P_2=[x_0,x_1],\\
&m_3=l_4-\tfrac{1}{2}l_2=x_0^2x_1-\tfrac{1}{2}x_1, && P_3=[x_0,[x_0,x_1]],\\
&m_4=l_7+l_5=x_0^2x_1^2+x_0x_1^2. && P_4=[x_0,[[x_0,x_1],x_1]].
\end{aligned}$$

The polynomials (m_j) must verify the condition (9)(ii), $m_4\rhd F_2=-\frac{1}{2}x_1$. We must find T such that $T\rhd F_2=\frac{1}{2}x_1$.

$$T=\tfrac{1}{2}x_1^2=\tfrac{1}{4}l_2^{\text{ш}2}.$$

$$\begin{aligned}
&Z_1=x_0,\\
&Z_2=x_0x_1+x_1,\\
&Z_3=x_0^2x_1-\tfrac{1}{2}x_1, \qquad \text{(9)(i)}\\
&Z_4=x_0^2x_1^2+x_0x_1^2+\tfrac{1}{2}x_1^2.
\end{aligned}$$

	Z_1	Z_2	Z_3	Z_4
P_1	1	0	0	0
P_2	0	1	0	0
P_3	0	0	1	0
P_4	0	0	0	1

The observation: By the algorithm in Fig. 1, we obtain

$$g=\tfrac{1}{2}Z_2^{\text{ш}2}-2Z_4\ \Rightarrow\ h=\tfrac{1}{2}z_2^2-2z_4.$$

The vector fields:

	$x_0\lhd$	$x_1\lhd$	
Z_1	1	0	
Z_2	0	$1+x_0$	$=1+Z_1$
Z_3	0	$-\frac{1}{2}+x_0^2$	$=-\frac{1}{2}+\frac{1}{2}Z_1^{\text{ш}2}$
Z_4	0	$x_0^2x_1+x_0x_1+\frac{1}{2}x_1$	$=Z_2+Z_3$

$$Y_0=\frac{\partial}{\partial z_1},$$

$$Y_1=(1+z_1)\frac{\partial}{\partial z_2}+(-\tfrac{1}{2}+\tfrac{1}{2}z_1^2)\frac{\partial}{\partial z_3}+(z_2+z_3)\frac{\partial}{\partial z_4}.$$

Those vector fields define the homomorphism $\mathcal{Y}$. Hence $(\mathcal{Y},h)$ is the differential representation of g.

References

[1] J. Berstel and C. Reutenauer, *Les Séries Rationnelles et leurs Langages* (Editions Masson, Paris, 1984).

[2] D'Alessandro, A. Isidori and A. Ruberti, Realisation and structure theory of bilinear dynamical systems, *SIAM J. Control* **12** (1974) 517-535.

[3] J.P. Duval, Génération d'une section des classes de conjugaison et arbre de mots de Lyndon de longueur bornée, Publication du LITP Paris (1988) 88-20.

[4] M. Fliess, Sur la réalisation des systèmes dynamiques bilinéaires, *C.R. Acad. Sci. Paris* **277** (série A) (1973) 923-926.

[5] M. Fliess, Sur divers produits de séries formelles, *Bull. Soc. Math. France* **102** (1974) 181-191.

[6] M. Fliess, Matrices de Hankel, *J. Math. Pure Appl.* **53** (1974) 197-222.

[7] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France* **109** (1981) 3-40.

[8] M. Fliess, Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices, *Invent. Math.* **71** (1983) 521-537.

[9] G. Jacob, Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives, in: *Séminaire d'Aussois, In Outils et Modèles mathématiques pour l'automatique, l'analyse des systèmes et le traitement du signal*, coord. I.D.LANDAU, éd. CNRS, 1-No. 567 (1980) 325-357.

[10] G. Jacob and N. Oussous, Sur un résultat de REE: séries de Lie et algèbres de mélange, Publication du LIFL Lille, IT-103, 1987.

[11] G. Jacob and N. Oussous, Un logiciel calculant la réalisation des systèmes analytiques de série génératrice finie, Publication du LIFL Lille, IT-123, 1988.

[12] G. Jacob and N. Oussous, Algebraic computation of analytical minimal realisation, Publication du LIFL Lille, IT-139, 1988.

[13] G. Jacob and N. Oussous, Réalisation locale et minimale des systèmes dynamiques non linéaires et mots de Lyndon, Publication du LIFL Lille, IT-140, 1988.

[14] B. Jakubczyk, Existence and uniqueness of realisations of nonlinear systems, *SIAM J. Control Optim.* **18** (1977) 728-740.

[15] R.E. Kalman, Mathematical description of linear dynamical systems, *SIAM J. Control* **1** (1963) 152-192.

[16] M. Lothaire, *Combinatorics on Words* (Reading, MA, 1983).

[17] *Macsyma Reference Manual*, Version Ten (Massachusetts Institute of Technology and Symbolics, 1983).

[18] G. Melançon and C. Reutenauer, Lyndon words, free algebras and shuffles, Publication du LITP Paris et UQAM Montreal, Quebec (1987) 87-63.

[19] D.E. Radford, A natural ring basis for the shuffle algebra and an application to group schemes, *J. Algebra* **58** (1979) 432-454.

[20] C. Reutenauer, The local realisation of generating series of finite Lie rank, in: M. Fliess and M. Hazewinkel, eds., *Algebraic and Geometric Methods in Nonlinear Control Theory* (1986) 33-43.

[21] H.J. Sussmann, Minimal realisations and canonical forms for bilinear systems, *J. Franklin Inst.* **301** (1976) 593-604.

[22] H.J. Sussmann, Existence and uniqueness of minimal realisations of nonlinear systems, *Math. Systems Theory* **10** (1977) 263-284.

[23] G. Viennot, *Algèbres de Lie libres et monoïdes libres*, Lecture Notes In Mathematics **691** (Springer, Berlin, 1978).